[SAG.A] Dokumentacja końcowa projektu
Protokół MAS z wykorzystaniem ontologii

Michał Aniserowicz <michalaniserowicz@gmail.com>
Jakub Turek <jkbturek@gmail.com>

2 czerwca 2013r.

1 Temat projektu

Tematem projektu jest implementacja protokołu wykorzystującego ontologię w systemie wieloagentowym. W ramach projektu opracowana została symulacja ruchu drogowego w kwadratowej sieci ulic przypominającej plan Manhattanu. Symulacja składa się z następujących elementów:

miasto określa ilość przecznic (wielkość przestrzeni),

kodeks ruchu drogowego definiuje zasady poruszania się na drodze,

sygnalizacja świetlna określa pierwszeństwo na części skrzyżowań,

samochód porusza się po mieście zgodnie z zasadami ruchu drogowego znajdującymi się w kodeksie.

Projekt obejmuje również przygotowanie graficznego interfejsu użytkownika, który pełni dwojaką funkcję:

- ukazuje aktualny stan miasta w rzucie z góry,
- pozwala na dynamiczną zmianę kodeksu ruchu drogowego.

2 Technologia

Projekt został zaimplementowany na platformie JADE[1]. Interfejs graficzny został stworzony z użyciem natywnych bibliotek języka Java (AWT oraz Swing). Implementacja tworzona była w środowisku Eclipse i testowana w systemie operacyjnym Windows 7 64-bit.

[1] Java Agent DEvelopment Framework - jade.tilab.org.

3 Agenci

W systemie zdefiniowanych jest pięć typów agentów:

Miasto określa rozmiar (w przecznicach) siatki miasta. Posiada zachowania umożliwiające sprawdzanie, w których kierunkach można przemierzyć skrzyżowanie w określonej lokalizacji. Zakłada się, że w jednym systemie symulacji występuje wyłącznie jeden agent miasta.

Kodeks ruchu drogowego określa następujące przepisy ruchu:

- stronę jezdni, którą poruszają się samochody,
- pierwszeństwo samochodu w zależności od jego typu,
- kolor sygnalizacji świetlnej oznaczający, że można wjechać na skrzyżowanie,
- wzajemne pierwszeństwo samochodów w sytuacji, gdy na skrzyżowaniu nie ma sygnalizacji świetlnej.

Zakłada się, że w jednym systemie symulacji występuje wyłącznie jeden agent kodeksu ruchu drogowego.

Samochód agent poruszający się po mieście. Posiada własną funkcję celu, która zmienia się wraz z upływem czasu. Komunikuje się z miastem, kodeksem ruchu drogowego, sygnalizatorami świetlnymi oraz innymi samochodami. Autonomicznie decyduje o swoim zachowaniu, to znaczy, że powinien (ale nie musi) respektować przepisy ruchu drogowego zdefiniowane w kodeksie. Porusza się w dziedzinie dyskretnej (skokowo), lecz w sposób płynny (wielostopniowe przejeżdżanie przez przecznicę). Występowanie agenta jest opcjonalne, dowolna jest liczba jego instancji w jednym systemie.

Sygnalizator świetlny steruje ruchem na skrzyżowaniu. Posiada dwa stany opisujące aktualny kolor (czerwony lub zielony), które ulegają cyklicznej zmianie. Reprezentuje grupę sygnalizatorów dla danego skrzyżowania (od dwóch do czterech, w zależności od liczby przecinających się na skrzyżowaniu jezdni). Występowanie agenta jest opcjonalne, dowolna jest liczba jego instancji w jednym systemie.

Mapa miasta odwzorowywuje, rzutem z góry, stan miasta w danej chwili czasu. Komunikuje się z samochodami oraz sygnalizatorami świetlnymi, odpytując cyklicznie stan agentów. Występowanie agenta jest opcjonalne, dowolna jest liczba jego instancji w jednym systemie.

4 Ontologia

Wykorzystanie ontologii umożliwiło opisanie zmiennych zasad ruchu drogowego, do których stosują się samochody. Ontologia jest wykorzystywana głównie przez trzy typy agentów:

Samochód nie zna autonomicznie topologii miasta. Stosuje się do obowiązujących zasad ruchu drogowego narzuconych przez kodeks.

Miasto posiada informację, w których kierunkach można przejechać przez dane skrzyżowanie.

Kodeks ruchu drogowego posiada informacje o obowiązujących przepisach ruchu drogowego.

4.1 Koncepcje

Tabela 1 przedstawia koncepcje zdefiniowane w ontologii ruchu drogowego.

Koncepcja	**Parametr**		
	Nazwa	**Typ**	**Dozwolone wartości**
Pozycja	Współrzędna x	Integer	$[0; city\_size)$
	Współrzędna y	Integer	$[0; city\_size)$
Kierunek	Kierunek	Integer	0 (północ), 1 (wschód), 2 (południe), 3 (zachód)
Samochód	Pozycja	Pozycja	-
	Kierunek	Kierunek	-
	Typ	Integer	0 (normalny), 1 (uprzywilejowany),
	Status	Integer	0 (jedzie), 1 (przed skrzyżowaniem), 2 (na skrzyżowaniu)
Sygnalizacja	Pozycja	Pozycja	-
	Kolor światła (północ)	Integer	0 (czerwone), 1 (zielone)
	Kolor światła (wschód)	Integer	0 (czerwone), 1 (zielone)
	Kolor światła (południe)	Integer	0 (czerwone), 1 (zielone)
	Kolor światła (zachód)	Integer	0 (czerwone), 1 (zielone)

Tabela 1: Koncepcje zdefiniowane w ontologii ruchu drogowego.

4.2 Predykaty

Samochody komunikują się z innymi agentami głównie przy użyciu predykatów. Tabela 2 przedstawia zdefiniowane w ontologii miasta predykaty.

Predykat	**Parametr**		
	Nazwa	**Typ**	**Dozwolone wartości**
Czy można skręcić?	Pozycja skrzyżowania	Pozycja	-
	Kierunek	Kierunek	-
Czy można przejechać?	Kolor światła	Integer	0 (czerwone), 1 (zielone)
Czy ma pierwszeństwo?	Samochód	Samochód	-
Czy ma pierwszeństwo?	Pozycja 1	Pozycja	-
	Pozycja 2	Pozycja	
Czy jeździ po stronie?	Strona	Integer	0 (lewa), 1 (prawa)

Tabela 2: Predykaty zdefiniowane w ontologii ruchu drogowego.

W dziedzinie predykatów, algorytm przejeżdżania przez skrzyżowanie można przedstawić następująco:

1. Sprawdź, w jakich kierunkach można pokonać skrzyżowanie. Prześlij do miasta predykat „czy można skręcić?" z niewiadomą typu kierunek, wypełniając jednocześnie pozycję skrzyżowania, przez które chcesz przejechać. Oczekuj na listę predykatów, które spełniają zadany warunek.

2. Sprawdź, czy Twój samochód posiada bezwzględne pierwszeństwo. Prześlij do kodeksu ruchu drogowego predykat „czy ma pierwszeństwo?" z pojedynczym parametrem i oczekuj na potwierdzenie jego prawdziwości.

   (a) Jeżeli predykat jest prawdziwy, wjedź na skrzyżowanie i zakończ algorytm.

   (b) Jeżeli predykat nie jest prawdziwy, kontynuuj algorytm.

3. Sprawdź, czy na skrzyżowaniu znajdują się światła. Jeżeli nie, kontynuuj algorytm od kolejnego punktu. Jeżeli tak, sprawdź kolor świateł i wyślij do kodeksu ruchu drogowego predykat „czy można przejechać?" uzupełniony kolorem światła. Oczekuj na potwierdzenie spełnienia predykatu.

(a) Jeżeli predykat jest prawdziwy, wjedź na skrzyżowanie i zakończ algorytm.

(b) Jeżeli predykat nie jest prawdziwy, kontynuuj algorytm.

4. Sprawdź, czy w pobliżu skrzyżowania znajdują się inne samochody. Jeżeli nie, wjedź na skrzyżowanie i zakończ algorytm. Jeżeli tak, dla każdego samochodu wypełnij predykat „czy ma pierwszeństwo?" z dwoma parametrami i oczekuj na potwierdzenie jego prawdziwości.

   (a) Jeżeli wszystkie predykaty są fałszywe, wjedź na skrzyżowanie i zakończ algorytm.

   (b) Jeżeli chociaż jeden z predykatów jest prawdziwy, oczekuj na przejechanie samochodu przez skrzyżowanie, a następnie powtórz algorytm od punktu 4.

5 Protokół

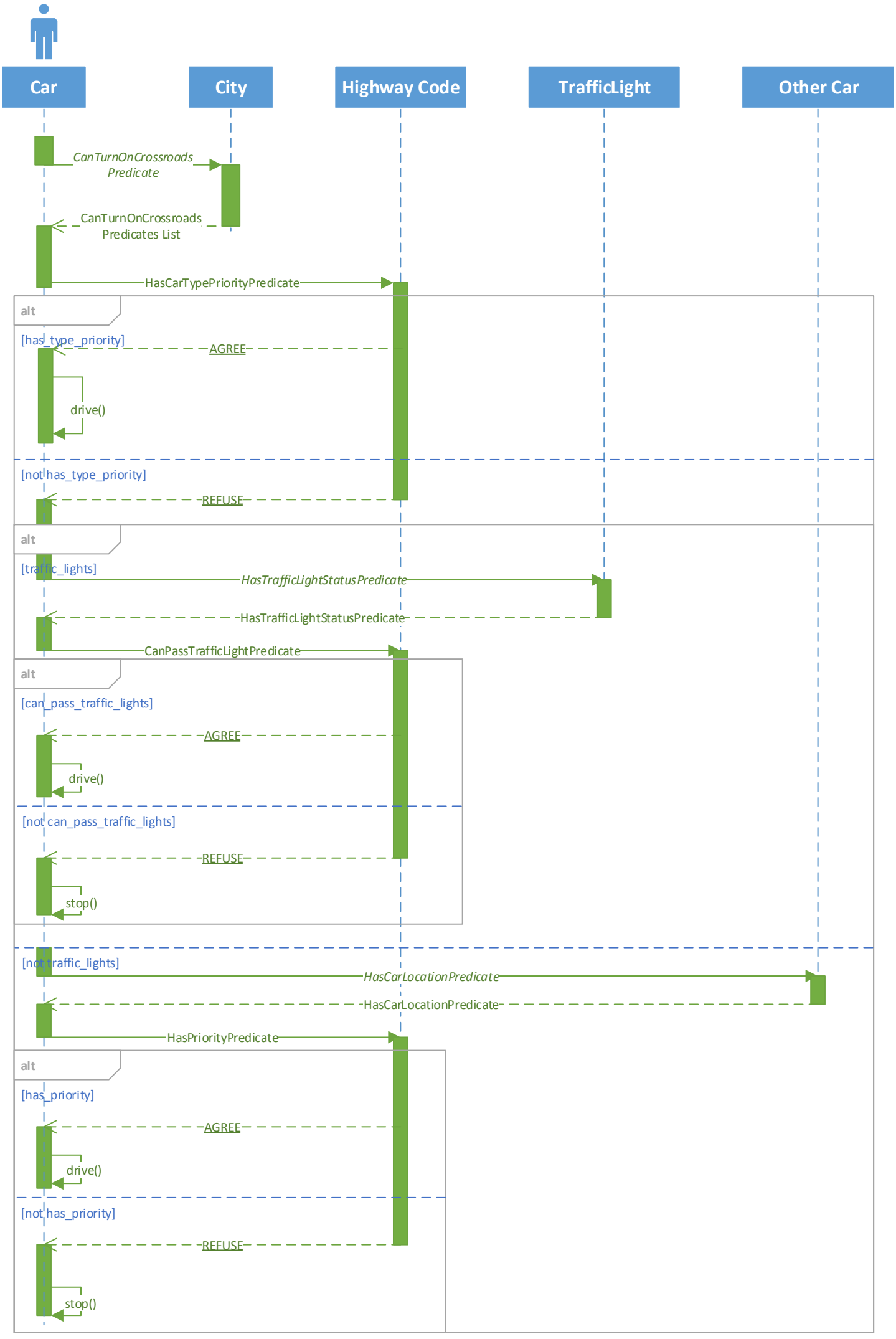
Wykorzystywany w symulacji ruchu ulicznego protokół został przybliżony razem z algorytmem opartym na predykatach w sekcji 4.2. Na następnej stronie przedstawiony jest jego większy fragment, który jest inicjowany ze strony samochodu przed wjazdem na skrzyżowanie.

Protokół jest w całości oparty na ontologii. Komunikacja polega na przesyłaniu komunikatów w postaci predykatów i oczekiwaniu na odpowiedź. Na diagramie widoczne są dwa różne scenariusze wymiany informacji:

- Przesłanie wypełnionego predykatu z akcją `QUERY_IF`. Oczekujemy na odpowiedź typu `INFORM_IF`, która zawiera wartość logiczną, będącą odpowiedzią na pytanie czy dany predykat jest prawdziwy, czy też nie. Scenariusz ten wykorzystywany jest przy odpytywaniu kodeksu drogowego o reguły dotyczące pierwszeństwa, wynikające z typu samochodu, statusu sygnalizacji świetlnej lub względnej pozycji dwóch samochodów.

- Przesłanie częściowo wypełnionego predykatu, z określonym polem zmiennej, z akcją `QUERY_REF`. Oczekujemy na odpowiedź, która zawiera predykat (lub listę predykatów) spełniający warunek zadany przez częściowo wypełnioną informację. Oczekiwana odpowiedź jest typu `INFORM_REF`.

Diagram nie uwzględnia komunikacji pomiędzy mapą oraz innymi agentami (odpytywanie o stan), a także fragmentu komunikacji pomiędzy samochodem i kodeksem ruchu drogowego dotyczącej informacji, którą stroną ulicy powinien poruszać się pojazd.


Car
City
Highway Code
TrafficLight
Other Car
CanTurnOnCrossroads Predicate
CanTurnOnCrossroads Predicates List
HasCarTypePriorityPredicate
alt
[has_type_priority]
AGREE
drive()
[not has_type_priority]
REFUSE
alt
[traffic_lights]
HasTrafficLightStatusPredicate
HasTrafficLightStatusPredicate
CanPassTrafficLightPredicate
alt
[can_pass_traffic_lights]
AGREE
drive()
[not can_pass_traffic_lights]
REFUSE
stop()
[not traffic_lights]
HasCarLocationPredicate
HasCarLocationPredicate
HasPriorityPredicate
alt
[has_priority]
AGREE
drive()
[not has_priority]
REFUSE
stop()

6 Parametry startowe agentów

W celu uruchomienia agentów należy podać właściwe im parametry startowe. W tabeli 3 zostało zaprezentowane ich zestawienie.

Agent	**Parametr**			
	Nazwa	**Typ**	**Dozwolone wartości**	**Wymagalność**
Samochód	Pozycja x	Integer	$[0; city\_size)$	Tak
	Pozycja y	Integer	$[0; city\_size)$	Tak
	Prędkość	Integer	$[0; \infty)$	Tak
	Kierunek	String	NORTH, EAST, SOUTH, WEST	Tak
	Typ	String	STANDARD, POLICE, AMBULANCE	Tak
	Kolor	String	Nazwy pól `java.awt.Color`	Nie
Sygnalizacja	Pozycja x	Integer	$[0; city\_size)$	Tak
	Pozycja y	Integer	$[0; city\_size)$	Tak
	Częstość zmiany	Integer	$[0; \infty)$	Tak
Miasto	Rozmiar	Integer	$[0; \infty)$	Tak

Tabela 3: Parametry startowe agentów.

Parametry startowe należy oddzielić od siebie przecinkami. Wymagane parametry należy podać w trakcie dołączania agenta do systemu (zarówno przez graficzny interfejs lub z poziomu konsoli). W przeciwnym razie rzucony zostanie wyjątek `InvalidAgentArgumentsException`.

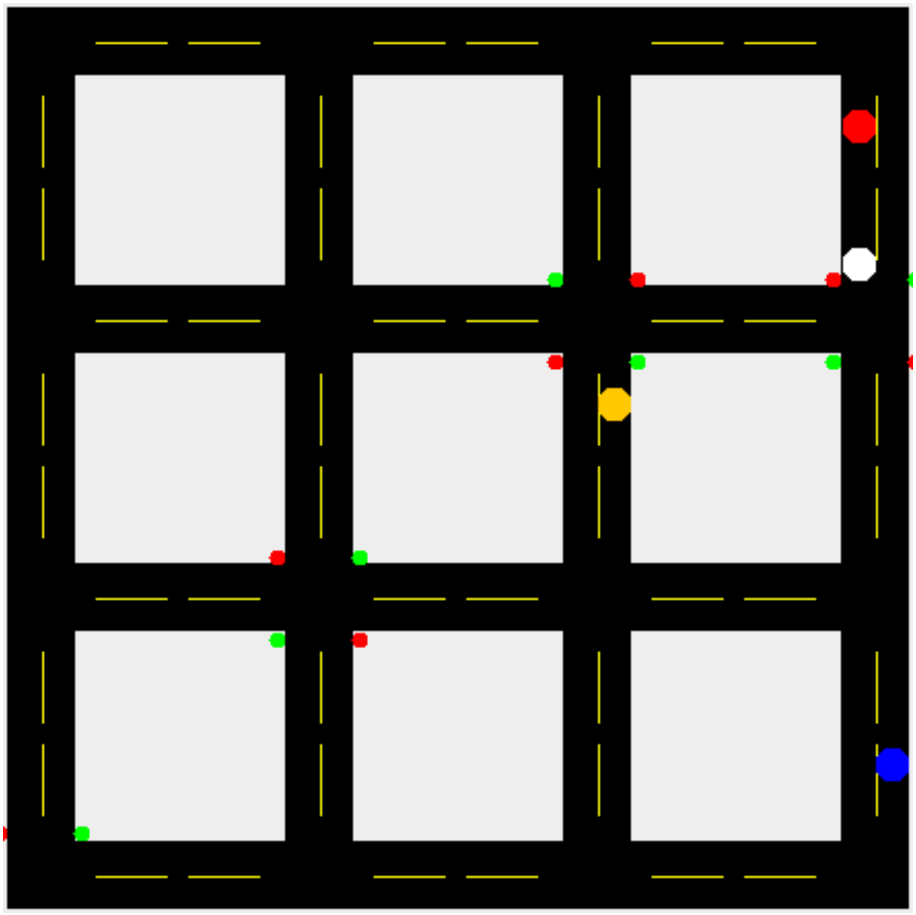
7 Graficzny interfejs użytkownika

Dwóch spośród wszystkich agentów zaimplementowanych w symulatorze posiada dedykowany interfejs graficzny, który ułatwia korzystanie z systemu:

Mapa Prezentuje stan agentów z perspektywy rzutu prostopadłego z nad miasta. Przykładowy widok interfejsu przedstawiony jest na rysunku 1. Różnokolorowe kropki znajdujące się na jezdni symbolizują samochody. Każdy samochód porusza się po odpowiednim pasie ruchu[2].

[2]W przykładzie prezentowany jest ruch prawostronny, co oznacza, że samochody pomarańczowy oraz niebieski poruszają się w kierunku północnym, natomiast biały i czerwony - południowym.

Sygnalizatory świetlne przedstawione są jako kropki ustawione tuż przed skrzyżowaniem, po stronie naturalnej dla wybranego typu ruchu[3]. Oznacza to, że dla przedstawionej na rysunku sytuacji (ruch prawostronny) pomarańczowy samochód ma zielone światło. Mapa przedstawia wierne odwzorowanie stanu obiektów, utworzone na podstawie informacji zwrotnej otrzymanej od tych obiektów. Oznacza to skokowy, proporcjonalny do ustawionej przez parametry startowe prędkości, ruch każdego pojazdu.

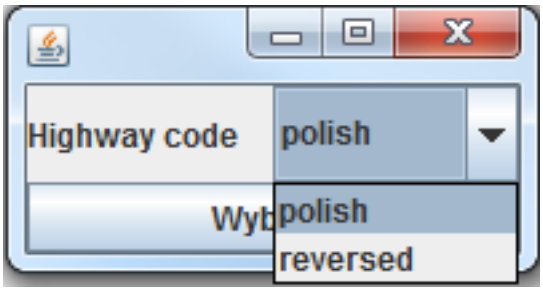


Rysunek 1: Graficzny interfejs użytkownika - widok mapy.

Kodeks ruchu drogowego Posiada prosty interfejs graficzny, na który składa się formularz posiadający jedno pole - rozwijaną listę dostępnych w symulacji kodeksów ruchu drogowego. Umożliwia przełączanie pomiędzy kodeksami w czasie działania programu. Wygląd formularza

[3]W przypadku ruchu prawostronnego po prawej stronie jezdni, w ruchu lewostronnym - po lewej.

przedstawia rysunek 2.



Rysunek 2: Graficzny interfejs użytkownika - wybór kodeksu ruchu drogowego.